```
************************
*                      *
* NASCOM EXTENDED BASIC *
*                      *
*     CP/M VERSION     *
*                      *
*       REV 3.2C       *
*                      *
************************
```

Rev 3.2 20 March 1983

Nascom Microcomputers
XBASIC IMPLEMENTATION NOTE

NASCOM EXTENDED BASIC REV 3.2

The version of Extended BASIC supplied for use with the Nascom Advanced Video Controller (AVC) uses features not described in the original AVC manual, Rev 1.1. There are also changes in the locations of routines and data. Therefore when using the AVC graphics version of Extended BASIC you should refer to versions of the AVC manual marked Rev 1.2 or later.

SOFTWARE REPORT FORM

PRODUCT: _____ SERIAL NUMBER: _____

DATE PURCHASED: _____

SUPPLIER: _____

REPORT BY: _____ DATE: _____

ADDRESS: _____

_____

_____

_____

TELEPHONE: _____

REPORT

Form SED17
Rev 1.0

NASCOM ENHANCED BASIC FOR CP/M

This manual describes the use of the Nascom Extended BASIC language. Extended BASIC (XBASIC) is available in three versions for use with tape, the NAS-DOS disc operating system and the CP/M disc operating system. The main body of this manual describes the general features of the language, and the Implementation Notes describe features peculiar to the CP/M version. We suggest a quick look at the Implementation Notes before you start reading the main manual. This will then enable you to note those areas of the main manual which differ from your CP/M version.

XBASIC IMPLEMENTATION NOTE

Please note the following errors and omissions in the Nascom Extended BASIC (XBASIC) Manual Rev 3.1T.

1. Chapter I, page 7
   In the last example in section 5.2, the first line should read:
   IF X>15 THEN A=1: ELSE A=0
   (i.e, the 1 and the 0 are interchanged).

2. Chapter I, page 8 and Appendix A, page 58. For '3/4', read '¾'.

3. Chapter II, page 9. The cursor movement keys described near the bottom of the page should be indicated by the appropriate arrows between the single quotes.

4. Chapter III, page 19. Near the top of the page, the assignment: LET AA=1+2*3/4 should, of course, assign the value 2.5 to AA, NOT 4.5.

5. Chapter III, page 24. In the POINT function description, the word CONJUCTION should, of course, read CONJUNCTION.

6. Chapter IV, page 30. In the example program given under the description of the SEP command, the variable MONTH should be changed to MNTH (note that MON is a reserved word).

7. Chapter IV, page 31. In the fourth example under FMT, the result of PRINT 7895 should display as 7.89500E+03, not as 7.895000E+03.

8. Chapter IV, page 33. The description of the ZONE command gives the parameters in the wrong order, and does not describe its use as a function. Here is the complete description of ZONE:

ZONE <J1>,<J2>
Sets the print zone (tab) width (<J2>), and the largest column for which printing to the next zone will stay on the same line (<J1>), known as the ZONE LIMIT. The default settings for these will vary according to the implementation, and will be found for your machine in the scratch-pad list at Appendix B, as WIDTHT (zone limit) and ZWIDTH (zone width). The current values for these may be obtained at any time by using ZONE as a function, i.e, PRINT ZONE(0) displays the zone limit, and PRINT ZONE(1) displays the zone width.

9. Chapter V, page 40. In the eighth line of section 6.2, change 9000 to 9030.

10. Chapter V, page 42. All references to "example b." should read "example 6.2"

11. Appendix C, page 72. In the description of the routine LEN1, location TYPE should read NTYPE.

12. Appendix D, page 76. In the listing of the HOME command, the location marked 4263: should read 4262:.

13. Appendix D, page 80. The default pointers for the auxiliary tables are incorrect. They should read as follows:

```
3A80: 00 42          Start of Auxiliary Reserved Word Table.
3A8A: 40 42          Start of Auxiliary Address Table.
```

To recap, and hopefully make this point clear - if extra words are to be entered as TEMPORARY additions, use the normal scratch-pad locations as in Appendix B. They will then be removed at the next 'cold start' to BASIC. If, on the other hand, they are to be PERMANENT additions, use the default locations given above. A 'cold start' will then still allow the use of your tables.

XBASIC CP/M IMPLEMENTATION NOTE

Please note that for the CP/M implementation of XBASIC the following should be substituted in place of the description given in the standard version of the manual.

II. THE SYSTEM EDITOR AND SYSTEM COMMANDS

1. SCREEN CONTROL CODES

The following screen control codes are used by XBASIC on the Nascom computer fitted with the Advanced Video Controller card, providing n 80 column by 25 line display.

Ctrl-A	&01	HOME cursor to top left corner of screen.
<TAB>	&09	TAB cursor to next print ZONE, by printing spaces. However, see also IOM command in Chapter IV.3.
<LF>	&0A	LINE FEED, or move cursor DOWN. Scroll screen at bottom.
<CS>	&0C	CLEAR SCREEN and Home cursor to top left corner.
<CR>	&0D	CARRIAGE RETURN, without line feed.
Ctrl-P	&10	PRINT SCREEN to printer (device £1, see Chapter IV.1).
'←'	&1C	Move cursor LEFT.
'→'	&1D	Move cursor RIGHT.
'↑'	&1E	Move cursor UP.
'↓'	&1F	Move cursor DOWN (same as <LF>).

2. THE XBASIC EDITOR

This powerful facility, available to you the moment that XBASIC is entered, has been designed in an attempt to make program entry and debugging a bit more of a pleasure rather than the pain which it becomes under most BASIC interpreters. Input lines may be up to 127 characters long, and note is kept at all times of where the start and finish of the line is. So, if you have several lines in a listing, you may move the cursor up the screen to that line and make modifications to it, even if it occupies two or more rows on the screen. If the line is extended so that it will apparently run into the next one, the lines below simply move down one row to make room for it. Note that the modified line is only entered into the program when the <CR> key is pressed while the cursor sits in one of the rows of the screen containing that line.

The following special key functions are available, the ones in brackets indicating the equivalents for the Nascom 1 keyboard:

Ctrl-A (@A)	HOME cursor to top left corner of screen.
s<BS> or <CS>	CLEAR screen and Home cursor.
'→'	Move cursor RIGHT.
'←'	Move cursor LEFT.
'↑'	Move cursor UP.
'↓'	Move cursor DOWN (scroll screen at bottom).
<BS>	DELETE character to the LEFT, but moving rest of line one place to the left.
S'<-' (@V)	DELETE character from the RIGHT, moving rest of line one place to the left.
S'->' (@W)	INSERT space at cursor, moving rest of line one place to the right, and moving lines below it one row down, if required. NOTE: An insertion done at the bottom line of the screen will cause an immediate scroll, moving the cursor up with it. This has no ill effects, apart from being a bit disconcerting when first observed (if you notice it).
Ctrl-Q (@Q)	ERASE whole line. This differs from Ctrl-X in that the cursor is returned to the start of the line before clearing it.
Ctrl-X (@X)	ERASE to end of line (even if it occupies 2 or more rows), from the current cursor position.
Ctrl-O (@O)	ERASE to end of screen from current cursor position.
Ctrl-P (@P)	PRINT SCREEN contents to printer.
<ESC> (S<NL>)	Abandons a line (though you could just use an arrow key or a Ctrl-Q!) and prints the 'Ok' prompt.
<CR> or <NL>	ENTER the current line on which the cursor sits into BASIC. The cursor will end up sitting at the start of the next line (i.e, not necessarily the next ROW of the screen). Leading and trailing spaces are ignored, and lines of greater than 127 characters will be truncated to 127 (this being the size of the buffer area).

The best way to become familiar with these functions is to use the system.

3. THE LINE EDITOR

In addition to the screen editor, a 'Line edit' mode is also available, primarily for use within programs, when to use the screen editor could cause some irritation (since the INPUT prompt would also be assumed to be part of the input line!! On the other hand, this could also be very useful in certain applications).

In this mode, cursor movement keys are not available, except that ' ' and <BS> both delete the last character from the line, Ctrl-P still gives a screen dump to printer, <ESC> abandons the line, and <CR> enters it into XBASIC.

For the reasons outlined above, screen edit mode is 'switched on' automatically in direct mode, and LINE EDIT mode turned on for programs. In addition, the user may use the IOM command (see

Chapter IV.3), inside or outside a program to change the editing mode: IOM 0,1 gives SCREEN EDIT mode, IOM 0,0 gives LINE EDIT mode. In direct mode, IOM 2,0 should be used before IOM 0,0 , otherwise screen edit mode will be reselected on completion of the statement.

LINE EDIT mode shows itself by means of a prompt at the start of the line (']' in direct mode and '?' in an INPUT statement with no specified prompt string).

SPECIAL NOTE: In spite of the declaration above that lines are limited to 127 characters in length, it is possible to move the buffer area to other areas in the memory space, and to change the buffer length up to 254 characters maximum! This may be done by means of the PTR command (see Chapter VII). Care must then be taken over selection of the area used to contain the buffer, and it is recommended that an area created by means of a CLEAR command be used.

4. SYSTEM COMMANDS

The follwoing commands are normally intended for use in direct mode, although some (such as RUN and LIST) can also be used to advantage within programs, and CHAIN is used almost entirely within programs. Because they all effect modification and control of programs and the system they are known as **system** commands.